Code motion for explicitly parallel programs

Full text Pdf (1.62 MB)

Source **Principles and Practice of Parallel Programming** archive
Proceedings of the seventh ACM SIGPLAN symposium on Principles and practice of parallel programming table of contents
Atlanta, Georgia, United States
Pages: 13 - 24
Year of Publication: 1999
ISSN:0362-1340
Also published in ...

Authors Jens Knoop
Bernhard Steffen

Sponsor SIGPLAN: ACM Special Interest Group on Programming Languages

Publisher ACM Press New York, NY, USA

↑ ABSTRACT

In comparison to automatic parallelization, which is thoroughly studied in the literature [31, 33], classical analyses and optimizations of explicitly parallel programs were more or less neglected. This may be due to the fact that naive adaptations of the sequential techniques fail [24], and their straightforward correct ones have unacceptable costs caused by the interleavings, which manifest the possible executions of a parallel program. Recently, however, we showed that unidirectional *bitvector analyses* can be performed for parallel programs as easily and as efficiently as for sequential ones [17], a necessary condition for the successful transfer of the classical optimizations to the parallel setting.In this article we focus on possible subsequent code motion transformations, which turn out to require much more care than originally conjectured [17]. Essentially, this is due to the fact that interleaving semantics, although being adequate for correctness considerations, fails when it comes to reasoning about *efficiency* of parallel programs. This deficiency, however, can be overcome by strengthening the specific treatment of synchronization points.

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

1 Jyh-Herng Chow , William Ludwell Harrison, III, Compile-time analysis of parallel programs that share memory, Proceedings of the 19th ACM SIGPLAN-SIGACT symposium on Principles of

h c g c cf c

programming languages, p.130-141, January 19-22, 1992, Albuquerque, New Mexico, United States

2 Patrick Cousot , Radhia Cousot, Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints, Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages, p.238-252, January 17-19, 1977, Los Angeles, California

3 P. Cousot and R. Cousot. Abstract interpretation frameworks. J. Logic and Computation, 2(4):511-547', 1992.

4 D. M. Dhamdhere, Practical adaption of the global optimization algorithm of Morel and Renvoise, ACM Transactions on Programming Languages and Systems (TOPLAS), v.13 n.2, p.291-294, April 1991

5 Matthew B. Dwyer , Lori A. Clarke, Data flow analysis for verifying properties of concurrent programs, ACM SIGSOFT Software Engineering Notes, v.19 n.5, p.62-75, Dec. 1994

6 Dirk Grunwald , Harini Srinivasan, Data flow equations for explicitly parallel programs, ACM SIGPLAN Notices, v.28 n.7, p.159-168, July 1993

7 Matthew S. Hecht, Flow Analysis of Computer Programs, Elsevier Science Inc., New York, NY, 1977

8 J. B. Kam and J. D. Ullman. Monotone data flow analysis frameworks. Acta Informatica, 7:305 - 317, 1977.

9 J. Knoop. Optimal Interprocedural Program Optimization: A new Framework and its Application. PhD thesis, Univ. of Kiel, Germany, 1993. LNCS Tutorial 1428, Springer-V., 1998.

10 Jens Knoop, Eliminating partially dead code in explicitly parallel programs, Theoretical Computer Science, v.196 n.1-2, p.365-393, April 6, 1998

11 J. Knoop. Parallel constant propagation. In Proc. 4th Europ. Conf. on Parallel Processing (Euro-Par'98), LNCS 1470, pages 445 - 455. Springer-V., 1998.

12 Jens Knoop , Oliver Rüthing , Bernhard Steffen, Lazy code motion, ACM SIGPLAN Notices, v.27 n.7, p.224-234, July 1992

13 J. Knoop, O. Riithing, and B. Steffen. Lazy strength reduction. J. Prog. Lang., 1(1):71-91, 1993.

14 Jens Knoop , Oliver Rüthing , Bernhard Steffen, Optimal code motion: theory and practice, ACM Transactions on Programming Languages and Systems (TOPLAS), v.16 n.4, p.1117-1155, July 1994

15 Jens Knoop , Oliver Rüthing , Bernhard Steffen, Partial dead code elimination, Proceedings of the ACM SIGPLAN 1994 conference on Programming language design and implementation, p.147-158, June 20-24, 1994, Orlando, Florida, United States

16 Jens Knoop , Oliver Rüthing , Bernhard Steffen, The power of assignment motion, ACM SIGPLAN Notices, v.30 n.6, p.233-245, June 1995

17 Jens Knoop , Bernhard Steffen , Jürgen Vollmer, Parallelism for free: efficient and optimal bitvector analyses for parallel programs, ACM Transactions on Programming Languages and Systems (TOPLAS), v.18 n.3, p.268-299, May 1996

h c g c cf c

18 A. Krishnamurthy and K. Yelick. Optimizing parallel SPMD programs. In Proc. 7th Int. Conf. on Lang. and Compilers for Parallel Comp. (LCPC'94), LNCS 892, pages 331 - 345. Springer- V., 1994.

19 Arvind Krishnamurthy , Katherine Yelick, Optimizing parallel programs with explicit synchronization, ACM SIGPLAN Notices, v.30 n.6, p.196-204, June 1995

20 L. Lamport. How to make a multiprocessor computer that correctly executes multiprocess programs. IEEE Trans. Comput., 28(9):690- 691, 1979.

21 J. Lee, S. P. Midkiff, and D. A. Padua. Concurrent static single assignment form and constant propagation for explicitly parallel programs. In Proc. I Oth Int. Conf. on Lang. and Compilers for Parallel Comp. (LCPC'97), pages 114- 130, 1997.

22 Douglas Long , Lori A. Clarke, Data flow analysis of concurrent systems that use the rendezvous model of synchronization, Proceedings of the symposium on Testing, analysis, and verification, p.21-35, October 08-10, 1991, Victoria, British Columbia, Canada

23 Kim Marriott, Frameworks for abstract interpretation, Acta Informatica, v.30 n.2, p.103-129, March 1993

24 S. P. Midkiff and D. A. Padua. Issues in the optimization of parallel programs, in Proc. Int. Conf. on Parallel Processing, Vol. II, pages 105- 113, 1990.

25 Steven S. Muchnick, Advanced compiler design and implementation, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1998

26 Steven S. Muchnick , Neil D. Jones, Program Flow Analysis: Theory and Application, Prentice Hall Professional Technical Reference, 1981

27 M. Sharir and A. Pnueli. Two approaches to interprocedural data flow analysis. In S. S. Muchnick and N. D. Jones, editors, Program Flow Analysis: Theory and Applications, chapter 7, pages 189-233. Prentice Hall, Englewood Cliffs, NJ, 1981.

28 Dennis Shasha , Marc Snir, Efficient and correct execution of parallel programs that share memory, ACM Transactions on Programming Languages and Systems (TOPLAS), v.10 n.2, p.282-312, April 1988

29 Harini Srinivasan , James Hook , Michael Wolfe, Static single assignment for explicitly parallel programs, Proceedings of the 20th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, p.260-272, March 1993, Charleston, South Carolina, United States

30 H. Srinivasan and M. Wolfe. Analyzing programs with explicit parallelism. In Proc. 4th Int. Conf. on Lang. and Compilers for Parallel Comp. (LCPC'91), LNCS 589, pages 405- 419. Springer- V., 1991.

31 M. Wolfe. High performance Compilers for Parallel Computing. Addison-Wesley, NY, 1996.

32 M. Wolfe and H. Srinivasan. Data structures for optimizing programs with explicit parallelism. In Proc. 1st Int. Conf. of the Austrian Center for Parallel Computation, LNCS 591, pages 139 - 156. Springer-V., 1991.

33 Hans Zima , Barbara Chapman, Supercompilers for parallel and vector computers, ACM Press, New York, NY, 1990

h c g c cf c

↑ CITINGS

Radu Rugina , Martin C. Rinard, Pointer analysis for structured parallel programs, ACM Transactions on Programming Languages and Systems (TOPLAS), v.25 n.1, p.70-116, January 2003

↑ INDEX TERMS

Primary Classification:
D. Software
↳ **D.1** PROGRAMMING TECHNIQUES

Additional Classification:
D. Software
↳ **D.3** PROGRAMMING LANGUAGES

F. Theory of Computation
↳ **F.3** LOGICS AND MEANINGS OF PROGRAMS

General Terms:
Algorithms, Design

Keywords:
bitvector problems, code motion (partial redundancy elimination), code optimization, data-flow analysis, interleaving semantics, shared memory, synchronization

↑ Collaborative Colleagues:

Jens Knoop:	Volker Braun Barbara M. Chapman Andreas Claßen Jean-Francois Collard Javier Esparza Alfons Geser G. Goos Manish Gupta Laurie J. Hendren Roy Dz-Ching Ju	Marion Klein J. Knoop Dirk Koschützki Gerald Lüttgen Gerald Luettgen Tiziana Margaria Eduard Mehofer Samuel P. Midkiff Michael F. P. O'Boyle Keshav Pingali	Oliver Rüthing Oliver Ruething Bernhard Steffen Jürgen Vollmer	
Bernhard Steffen:	Michael von der Beeck Volker Braun Ed Brinksma Georg Brune Olaf Burkart Didier Caucal Andreas Claßen Rance Cleaveland A. Dannecker Burkhard Freitag C. Friedrich Alfons Geser	Werner Goerigk Susanne Graf Andreas Hagerer Hardi Hungar Hans-Dieter Ide Anna Ingólfsdóttir Bengt Jonsson Marion Klein Jens Knoop Dirk Koschützki Jürgen Kreileder Gerald Lüttgen Kim G. Larsen	Kim Guldstrand Larsen Gerald Luettgen Markus Müller-Olm Tiziana Margaria Gustaf Naeser Oliver Niese Rita Nisius Jan Nyström Ernst-Rüdiger Olderog Joachim Parrow Joachim Posegga	Manfred Reitenspieß Oliver Ruething David A. Schmidt F. Schreiber Helmut Seidl Scott A. Smolka Rob J. VanGlabbeek Jürgen Vollmer Carsten Weise Haiseung Yoo Ulrich Zukowski

h c g c cf c

Rob J. Glabbeek

Roland Rückert
Oliver Rüthing

h c g c cf c

IEEE 4-22-2004

Your search matched **32** of **1027552** documents.
A maximum of **500** results are displayed, **50** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

convert*<and>parallel*<and>branch* Search

☐ Check to search within this result set

h eee e eee g e ch e ch e e c e c e e

IEEE 4-22-2004

17 **Profile-driven instruction level parallel scheduling with application to super blocks**

Chekuri, C.; Johnson, R.; Motwani, R.; Natarajan, B.; Ran, B.R.; Schlansker, M.;
Microarchitecture, 1996. MICRO-29. Proceedings of the 29th Annual IEEE/ACM International Symposium on , 2-4 Dec. 1996
Pages:58 - 67

[Abstract] [PDF Full-Text (856 KB)] IEEE CNF

h eee e eee g e ch e ch e e c e c e e